

Exercices tirés du livre de V.P. Heuring and H.F. Jordan  
**Computer Systems Design and Architecture, 1997.**

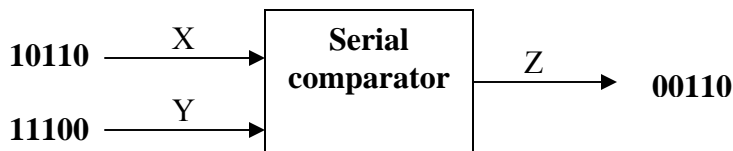
Circuits Logiques Séquentiels: A.34; A.35; A.38; Compteur Code Gray modulo 8.

**A.34**

Design an FSM that outputs a 1 when the last three inputs are 011 or 110. Just show the state table. Do not draw a circuit.

**A.35**

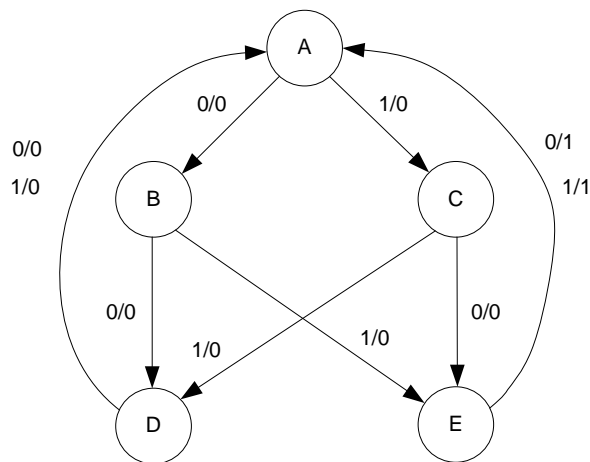
Design a finite state machine that takes two binary words X and Y in serial form, least significant bit (LSB) first, and produces a 1-bit output Z that is true when  $X > Y$  and is 0 for  $X \leq Y$ . When the machine starts, assume that  $X = Y$ . That is, Z produces 0's until  $X > Y$ . The diagram shows a sample input sequence and the corresponding output sequence.



**A.38**

Given the state transition diagram shown below, do the following:

- Create a state table.
- Design a circuit for the state machine described by your state table, using D flip-flop(s), a single decoder, and OR gates. For the state assignment, use the bit pattern that corresponds to the position of each letter in the alphabet, starting from 0. For example, A is at position 0, so the state assignment is 000; B is at position 1, so the state assignment is 001, and so on.



### Code Gray modulo 8 Counter

Let us use the FSM design procedure to design a synchronous 3-bit Gray code counter using positive edge-triggered D flip-flops. A gray code is a code in which consecutive numbers differ in only one bit. Thus a 3-bit Gray code consists of the following sequence: 000, 001, 011, 010, 110, 111, 101, 100. The resulting state diagram is shown in Figure 1a. The set of eight states for this state diagram can be encoded simply by using the Gray code output values as the state variables. The resulting state transition table with this encoding is shown in Figure 1b.

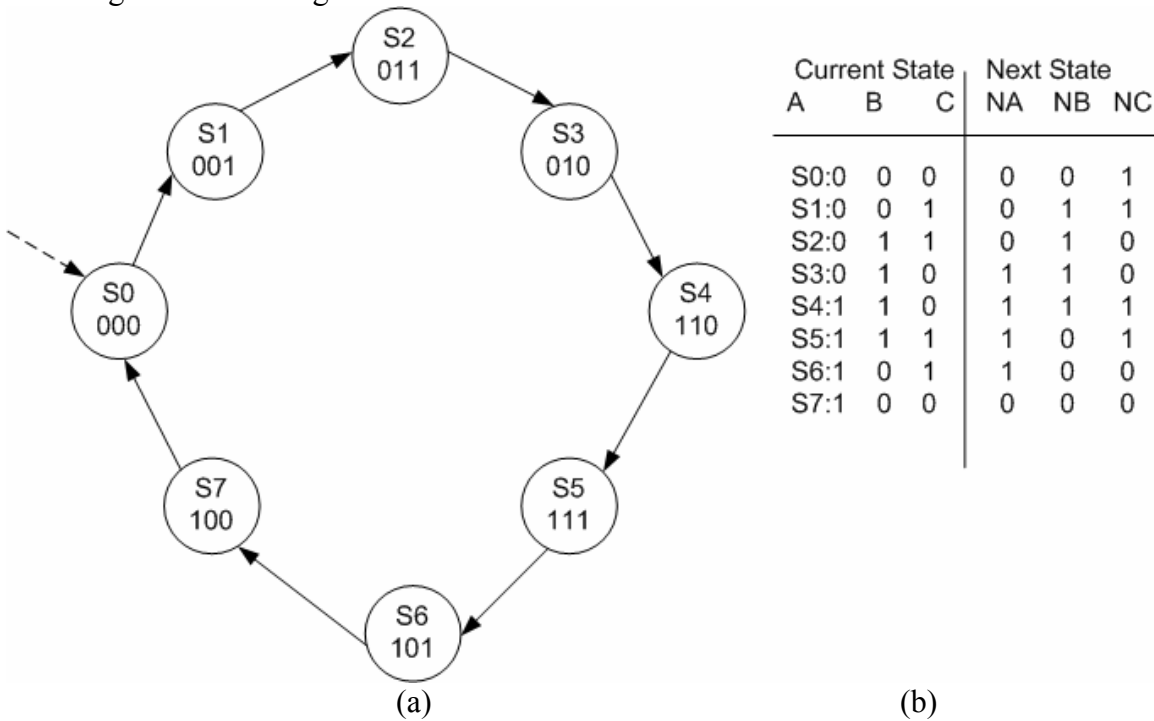


Figure 1

Find the logic diagram for a 3-bit Gray code counter.